African Masters of Machine Intelligence
Foundation of Machine Learning
Dr. Moustapha Cisse

1 Course Content, Resource Materials & Introduction

1.1 Course Content

1. Basic & foundation knowledge
    - Introduction to Linear Algebra
    - Fundamental concepts of Probability and Statistics
    - Introduction to Programming with Python
2. Foundation of Machine Learning
    - General Introduction to Machine Intelligence
    - Regression Algorithm - Linear Regression
    - The Concept of Maximum Likelihoood Estimation
    - Classification Algorithm - Logistic Regression
    - K-Fold Cross Validation
    - Feature Selection

1.2 Resource Materials

- Pattern Recognition & Machine Learning
- Deep Learning
- Understanding Machine Learning from Theories to Algorithms

2 General Introduction to Foundation of Machine Learning

The conventional approach to machine learning pipeline can be simply pictured with following arrow diagrams:

$$\text{Data}\{x_i, y_i\}_{i=1}^{n} \Longrightarrow \text{Hypothesis} \Longrightarrow \text{Criterion} \Longrightarrow \text{Learning Algorithms} \Longrightarrow \text{Hypothesis} \Longrightarrow \text{Predictions}$$

Mathematically,

$$h : \mathbb{E} \to \mathbb{F}$$

If we take $\mathbb{E} = \mathbb{R}^d$ and $\mathbb{F} = \mathbb{R}^c$, then d: number of features/attributes in the training dataset and c: number of targets in the testing dataset.

3 Supervised Learning

3.1 Regression: Linear Regression

Here, we considered a classical example of regression algorithm called Linear Regression defined as

$$h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

where θ_i's are the parameters(weights) parameterizing the space of linear functions mapping from $\mathcal{X}$ to $\mathcal{Y}$. For simplicity, we take the intercept term: $x_0 = 1$ so that

$$h_\theta = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

Loss function: $l(\theta) = \frac{1}{2} \sum_{i=0}^{n} (h_\theta - y_i)^2$

Objective function: $\min_\theta l(\theta) = \sum_{i=0}^{n} (h_\theta - y_i)$ θ_0 : initial solution

Repeat: $\theta_j : \theta_{j-1} - \alpha \frac{\partial}{\partial \theta_{j-1}} l(\theta_{j-1})$ α : learning rate

Repeat until convergence: $\theta_j : \theta_{j-1} - \alpha (y^i - h_\theta(x_j^i)) x_j^i$

$\theta_j : \theta_{j-1} - \alpha \sum_{i=0}^{n} (y_i - h_\theta(x_i)) x_i$ **Batch Gradient Descent**

$\theta : \theta - \alpha (y_i - h_\theta(x_i)) x_i$ **Stochastic Gradient Descent**

Remark 1

- *The choice of the magnitude of α matters as its high magnitude will result in overshooting i.e oscillating around the solution of the objective function and likewise its small magnitude will require a large number iterations to reach the objective function.*
- *The batch gradient descent is computationally costly and high complexity as it requires iterating over each given data points of the training dataset in updating the parameter θ. Hence, the preference of the stochastic gradient descent over the batch gradient descent since it only picks a particular data point.*
- *However, the stochastic gradient descent is highly subjected to noise thereby increasing the variance of the model. Therefore, to strike a balance, expert developed a merger of both by making the choice of the number data points p such that $n \geq p$. This is called the **Mini Batch Gradient Descent***

$$\theta_j : \theta_{j-1} - \alpha \sum_{i=0}^{p} (y_i - h_\theta(x_i)) x_i$$

3.2 Derivation of Normal Equation Using Matrix Notation

We will represent the loss function in vector and matrix notation and proceed with its derivation of the normal equation as follows:

$$\begin{aligned} l(\theta) &= \frac{1}{2}\|X\theta - y\|^2 = \frac{1}{2}(X\theta - y)^T(X\theta - y) \\ &= \frac{1}{2}(\theta^T X^T - y^T)(X\theta - y) = \frac{1}{2}(\theta^T X^T X\theta - \theta^T X^T y - y^T X\theta + yy^T) \\ &= \frac{1}{2}(\theta^T X^T X\theta - 2y^T X\theta + yy^T) \qquad \text{Since } \theta^T\theta = \theta^2 \\ \nabla_\theta &= \frac{1}{2}(2X^T X\theta - 2y^T X) \qquad \text{Then, } \min_\theta l(\theta) = \nabla_\theta = 0 \\ &\frac{1}{2}(2X^T X\theta - 2y^T X) = 0 \implies X^T X\theta = y^T X \end{aligned}$$

$$\boxed{\theta = (\mathbf{XTX})^{-1}\mathbf{y^T X}} \tag{1}$$

Remark 2 *We claim the solution in 1 if and only if $X^T X$ is invertible, otherwise an application of its pseudo-inverse or penalizing the parameter θ_i i.e assuming θ_i follows the guassian distribution. A more emphasis on will stressed in later section.*

4 The Concept of Maximum Likelihood Estimator

To proceed, we consider the following assumptions:- Given

$$y_i = \theta_i x_i + \epsilon_i$$

with

- $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$
- ϵ_i is independently and identically distributed (i.i.d)
- θ_i is not a random variable

Then, the probability density function is given as

$$\begin{aligned} \mathbb{P}(\epsilon_i) =& \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-\epsilon_i}{2\sigma^2}\right) \\ =& \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y_i - h_\theta(x_i))^2}{2\sigma^2}\right) \end{aligned}$$

Since ϵ_i is i.i.d, then the *likelihood* of the target given the data is defined as

$$\mathbb{P}(y_i|x_i;\theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y_i - h_\theta(x_i))^2}{2\sigma^2}\right) \tag{2}$$

Definition 1 (Maximum Likelihood Estimation) *The maximum likelihood estimation is the parameter that maximizes the probability of a given data.*

Taking the maximum of the log-likelihood of 2, we have

$$\begin{aligned}\max_{\theta} \log l(\theta) &= \max_{\theta} \log \left[\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y_i - h_\theta(x_i))^2}{2\sigma^2} \right) \right] \\ &= \max_{\theta} \sum_{i=1}^{n} - \left(\frac{y_i - h_\theta(x_i)}{2\sigma^2} \right) + \text{constant term} \\ &= \min_{\theta} \sum_{i=1}^{n} \left(\frac{y_i - h_\theta(x_i)}{2\sigma^2} \right) = \min_{\theta} l(\theta) \end{aligned} \tag{3}$$

5 Classification

Given $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}, y_i \in \mathbb{R}$ if $y_i \in \{0, 1\}$ or $\{-1, +1\}$ and $h_\theta(x) = \dfrac{1}{1 + e^{-\theta^T X}}$ where $g(x) = \dfrac{1}{1 + e^{-x}}$. We assume that

$$\begin{aligned}\mathbb{P}(y = 1|x; \theta) &= h_\theta(x) \\ \mathbb{P}(y = 0|x; \theta) &= 1 - h_\theta(x) \\ \mathbb{P}(y|x; \theta) &= h_\theta(x)^y \left(1 - h_\theta(x)\right)^{1-y}\end{aligned}$$

According to MLE $l(\theta) = \mathbb{P}(y|x : \theta)$ on a single data point, considering the fact that the data points are independently and identically distributed then the loss,

$$\begin{aligned} l(\theta) &= \prod_{i=1}^{n} \mathbb{P}\left(y_i|x_i : \theta\right) \\ &= \prod_{i=1}^{n} h_\theta(x)^y \left(1 - h_\theta(x)\right)^{1-y} \\ {}^* \log\left(l(\theta)\right) &= \sum_{i=0}^{n} \Big[y_i \log h_\theta(x_i) + (1 - y_i) \log(1 - h_\theta(x_i)) \Big] \qquad \text{log-likelihood} \end{aligned}$$

At the heart of gradient descent

$$\theta_i = \theta + \alpha \nabla_\theta l(\theta)$$

Recall $h_\theta = g(\theta^T x) = \dfrac{1}{1 + e^{(-\theta^T x)}}$

$$\begin{aligned} \frac{\partial}{\partial \theta_i} l(\theta) &= \Big(y \frac{1}{g(\theta^T x)} - (1 - y) \frac{1}{g(\theta^T x)} \Big) \frac{\partial}{\partial \theta} g(\theta^T x) \\ &= \Big(y \frac{1}{g(\theta^T x)} - (1 - y) \frac{1}{g(\theta^T x)} \Big) g(\theta^T x)(1 - g(\theta^T x)) \frac{\partial}{\partial \theta} \theta^T x \\ &= \big[y(1 - g(\theta^T x) - (1 - y) g(\theta^T x)) \big] \frac{\partial}{\partial \theta} \theta^T x \\ &= \big[y(1 - g(\theta^T x)) - (1 - y) g(\theta^T x) \big] x^i = \Big[y - g(\theta^T x) \Big] = \Big(y - h_\theta(x) \Big) x^i \end{aligned}$$

Therefore, the stochastic gradient descent is given as:
Repeat until convergence:

$$\boxed{\theta_j; \theta_j + \alpha(y_i - h_\theta(x_i)) x_j^i}$$

Corrollary 1 (The Notion of Empirical Risk Minimization) *Given that* $\mathbb{P}(x, y)$ *i.e* $(x_i, y_i) \sim \mathbb{P}$ *with defined loss function* $l(\theta) = \dfrac{1}{2}(h_\theta(x) - y_i)^2$*, then the (true) Risk Minimization is defined as*

$$\mathbb{R}_{h\theta} = \min_{\theta} \mathbb{E}_{(x,y)\sim\mathbb{P}} (h_\theta(x) - y_i)^2$$

and the Empirical Risk Minimization is defined as

$$\min_{\theta} \sum_{i=1}^{n} (h_\theta(x) - y_i)^2$$

Assuming the linear regression model $y = h^*_\theta(x) + \epsilon_i, \epsilon_i \sim \mathcal{N}(0, \sigma^2)$, formally we decompose the true risk to

$$\begin{aligned}
\mathbb{R}_{\mathrm{h}\theta} &= \mathbb{E}\Big[(y - h_\theta(x))^2\Big] \\
&= \mathbb{E}\Big[(h^*_\theta + \epsilon_i - h_\theta(x))^2\Big] \\
&= \mathbb{E}(\epsilon_i^2) + \mathbb{E}\Big[2\epsilon_i(h^*_\theta(x) - h_\theta(x))\Big] + \mathbb{E}\Big[\big(h^*_\theta(x) - h_\theta(x)\big)^2\Big] \\
&= \sigma^2 + \mathbb{E}\Big[\big(h^*_\theta(x) - h_\theta(x)\big)\Big]^2 + \mathrm{Var}\big(h^*_\theta(x) - h_\theta(x)\big)
\end{aligned}$$

where $\sigma^2 -$ Noise, $\mathbb{E}\Big[\big(h^*_\theta(x) - h_\theta(x)\big)\Big]^2 -$ Bias, $\mathrm{Var}\big(h^*_\theta(x) - h_\theta(x)\big) -$ Variance

Remark 3

- **Maximizing the likelihood is equivalent to maximizing the log -likelihood (strictly increasing function) which is also equivalent to minimizing the negative log-likelihood*
- *The loss function (cross entropy) is the measure of degree of disorderliness or misinformation in the data*
- *Ideally, our choice is to minimize the risk for the population i.e the true risk with high σ^2 accounting for the randomness in the data, high bias indicating under-fitting (poor performance on both the training and testing dataset) and high variance indicating over-fitting (good performance on the training dataset)*

6 K-Fold Cross Validation

In order to see to the efficiency of the trade-off of bias and variance, we implement the *k-fold cross validation* algorithms as follows:

- Randomly split D into k disjoint subsets of size n

$$D = \bigcup_{i=1}^{k} D_i$$

- For each model M_i:
  - For each $j = 1, \ldots, \ldots k$:
    - Train M_i on $D - D_i$
    - Test M_{ij} on $D_i \to l_{Dj}(M_{ij})$
- Generalization error of M_i = Average of $l_{Dj}(M_{ij})$

7 Feature Selection

Suppose we have $d-$ features with 2^d-subset of features then we present the two types of feature selection:-

7.1 Wrapper Methods

- Initialize $F = \phi$
- Repeat: {
  1. For i in $range(d)$:
     - if $i \notin F$, let $F_i = F \bigcup \{i\}$
     - Use cv to evaluate F_i
  2. set F to be the best subset found at step 1.
- Select the best subset that was found doing the entire procedure

7.2 Filter Methods

This approach examine the correlation between x^i and y

$$\mathcal{MI}(x^{(i)}, y) = \mathcal{K}\text{l}\Big(\mathbb{P}(x^{(i)}, y) \parallel \mathbb{P}(x^{(i)})\mathbb{P}(y)\Big)$$

$\mathcal{MI}$:Mutual Information, $\mathcal{K}$l :Kulback Liebler Divergence

Remark 4
- *The algorithm complexity for wrapper and filter methods are $\mathcal{O}(d^2)$ and $\mathcal{O}(d)$ respectively.*
- *Highly feature correlations does not necessarily implies feature co-linearity*